CLAIMS

1. A network interface adapter, comprising:

a network interface, coupled to send and receive data packets over a network;

a host interface, for coupling to a host processor and to a system memory associated therewith, the system memory containing context information with respect to a plurality of transport service instances used to send and receive the data packets over the network, each of the data packets belonging to a respective one of the service instances;

packet processing circuitry, coupled between the network interface and the host interface, and adapted to process the data packets using the context information of the respective service instances; and

a cache memory associated with the packet processing circuitry and coupled to load from the system memory and store the context information of the respective transport service instances for the data packets being processed by the packet processing circuitry.

2. An adapter according to claim 1, wherein the transport services instances have respective instance numbers, and wherein the cache memory comprises one or more tables having entries indicating the context information of the respective transport service instances, indexed by a portion of the service instance numbers.

3. An adapter according to claim 2, wherein the portion of the instance numbers comprises a predetermined number of the least significant bits of the instance numbers.

4. An adapter according to claim 2, wherein the one or more tables comprise at least two tables.

5. An adapter according to claim 2, wherein the entries comprise respective target fields, corresponding to at least a segment of the service instance numbers of the service instances to which the entries belong, and wherein the target fields are compared to the segment of the service instance numbers of the data packets to determine that a cache hit has occurred, whereupon the packet processing circuitry reads the context information from one of the tables.

6. An adapter according to claim 5, wherein when the cache hit does not occur, the context information is read from the system memory and loaded into the cache memory.

7. An adapter according to claim 1, wherein the packet processing circuitry comprises a cache controller, which is adapted, responsive to a request from the circuitry to access the context information in the cache memory with respect to one of the service instances, to determine whether a cache hit has occurred, and when the cache hit has not occurred, to read the requested context information from the system memory and load the requested context information into the cache memory in place of the context information of another one of the service instances.

8. An adapter according to claim 7, wherein the context information is organized in the cache memory using a plurality of tables having entries referenced by respective indices, and wherein the cache controller is adapted, while reading the requested context information from the system memory for one of the service instances

having a given one of the indices, to block access by the packet processing circuitry to the context information of the service instances having the given one of the indices, while enabling the packet processing circuitry to access the context information of the service instances with other indices.

9. An adapter according to claim 7, wherein the cache controller is adapted, responsive to the request to access the context information, to set a flag with respect to the service instance for which the context information is requested indicating that the context information is in use, and wherein the cache controller is further adapted, upon loading the context information into the cache memory, to store the loaded context information in the cache in place of the context information of another one of the service instances whose flag is not set.

10. An adapter according to claim 1, wherein the context information loaded into the cache memory comprises one or more fields that are updated by the packet processing circuitry in the course of processing the data packets, and wherein the updated fields are copied back to the context information in the system memory after the data packets have been processed.

11. An adapter according to claim 10, wherein the updated fields comprise packet serial numbers of packets processed by the circuitry.

12. An adapter according to claim 1, wherein the context information stored in the cache memory comprises a send cache, containing the context information pertaining to packets generated responsive to requests submitted by the

host processor, and a receive cache, containing the context information pertaining to packets generated responsive to requests submitted to the adapter by remote entities over the network.

13. An adapter according to claim 1, wherein the packet processing circuitry comprises:

an outgoing packet generator, adapted to generate the packets for delivery to remote entities via the network; and

an incoming packet processor, coupled to receive and process the packets from the remote entities via the network,

wherein both the outgoing packet generator and the incoming packet processor are coupled to access the same context information in the cache memory.

14. An adapter according to claim 13, wherein the outgoing packet generator is adapted to generate the packets for delivery to the remote entities responsive both to outgoing requests submitted by the host processor via the host interface and to incoming requests conveyed by the packets received from the remote entities.

15. An adapter according to claim 14, wherein the incoming packet processor is adapted to process both the packets that are received from the remote entities responsive to the outgoing requests conveyed by the packets delivered to the remote entities and the packets that are received from the remote entities conveying the incoming requests.

16. An adapter according to claim 1, wherein the transport service instances comprises queue pairs, which

are used to interact with a transport layer of the network.

17. A method for network communications, comprising:

coupling a network adapter having a cache memory between a host processor and a network;

storing context information in a system memory associated with the host processor, the context information relating to a plurality of transport service instances for use in sending and receiving data packets over the network via the adapter, each of the data packets belonging to a respective one of the service instances;

responsive to the use of a subset of the transport service instances to send and receive the data packets, loading into the cache memory the context information relating to the transport service instances in the subset; and

processing the data packets using the adapter based on the context information in the cache memory.

18. A method according to claim 17, wherein the transport services instances have respective instance numbers, and wherein loading the context information into the cache memory comprises using respective entries in one or more tables in the cache memory to provide access to the context information, wherein the entries are indexed by a portion of the service instance numbers.

19. A method according to claim 18, wherein the portion of the instance numbers comprises a predetermined number of the least significant bits of the instance numbers.

20. A method according to claim 18, wherein storing the context information in the one or more tables comprises storing the context information in at least two tables.

21. A method according to claim 18, wherein storing the context information comprises storing at least a segment of the respective service instance numbers of the service instances to which the entries belong in respective target fields of the entries, and wherein processing the data packets comprises comparing the target fields to the segment of the service instance numbers of the data packets to determine that a cache hit has occurred, whereupon the packet processing circuitry reads the context information from one of the tables.

22. A method according to claim 21, wherein loading the context information comprises, when the cache hit does not occur, reading the context information from the system memory and loading the information read from the system memory into the cache memory.

23. A method according to claim 17, wherein loading the context information into the cache memory comprises receiving a request to access the context information in the cache memory with respect to one of the service instances, and upon determining that a cache hit has not occurred, reading the requested context information from the system memory, and storing the requested context information into the cache memory in place of the context information of another one of the service instances.

24. An adapter according to claim 23, loading the context information into the cache memory comprises organizing the context information in the cache memory using a plurality of tables having entries referenced by

respective indices, and comprising, while reading the context information from the system memory for one of the service instances having a given one of the indices, blocking access to the context information of the service instances having the given one of the indices, while allowing access the context information of the service instances with other indices.

25. A method according to claim 23, wherein loading the context information read from the system memory into the cache memory comprises setting a flag with respect to the service instance for which the context information is requested indicating that the context information is in use, and wherein storing the requested context information comprises storing the requested context information in place of the context information of another one of the service instances whose flag is not set.

26. A method according to claim 17, wherein processing the data packets comprises updating one or more fields of the context information in the cache memory, and comprising copying the updated fields back to the context information in the system memory after the data packets have been processed.

27. A method according to claim 26, wherein updating the one or more fields comprises writing to the cache memory packet serial numbers of packets processed by the circuitry.

28. A method according to claim 17, wherein loading the context information comprises loading into a send cache the context information pertaining to packets generated responsive to requests submitted by the host processor,

and loading into a receive cache the context information pertaining to packets generated responsive to requests submitted to the adapter by remote entities over the network.

29. A method according to claim 17, wherein processing the data packets comprises both generating some of the packets for delivery to remote entities via the network and processing others of the packets received from the remote entities via the network, using the same context information in the cache memory.

30. A method according to claim 29, wherein generating the packets comprises generating the packets responsive both to outgoing requests submitted by the host processor and to incoming requests conveyed by the packets received from the remote entities, using the context information in the cache memory.

31. A method according to claim 30, wherein processing the packets that are received from the remote entities comprises receiving and processing both the packets returned by the remote entities responsive to the outgoing requests in the packets delivered to the remote entities and the packets that are received from the remote entities conveying the incoming requests, using the context information in the cache memory.

32. A method according to claim 17, wherein the transport service instances comprises queue pairs, which are used to interact with a transport layer of the network.

33. A host channel adapter, comprising:

709/213

a fabric interface, coupled to send and receive data packets over a switch fabric;

a host interface, for coupling to a host processor and to a system memory associated therewith, the system memory containing context information with respect to a plurality of queue pairs, each of the data packets belonging to a respective one of the queue pairs;

packet processing circuitry, coupled between the fabric interface and the host interface, and adapted to process the data packets using the context information of the respective queue pairs; and

a cache memory associated with the packet processing circuitry and coupled to load from the system memory and store the context information of the respective queue pairs for the data packets being processed by the packet processing circuitry.

34. An adapter according to claim 33, wherein the packet processing circuitry comprises:

an execution unit, adapted to generate the packets for delivery to remote entities via the network; and

a transport check unit, coupled to process the packets received from the remote entities via the network,

wherein both the execution unit and the transport check unit are coupled to access the same context information in the cache memory.

35. An adapter according to claim 34, wherein the execution unit is adapted to generate the packets for delivery to the remote entities responsive both to work queue entries submitted by the host processor via the host interface and to incoming requests conveyed by the

transport check unit to the execution unit, in response to the packets received from the remote entities.

36. An adapter according to claim 35, wherein the transport check unit is adapted to process both the packets that are received from the remote entities in response to the packets sent over the network responsive to the work queue items and the packets that are received from the remote entities conveying the incoming requests.

37. A method for network communications, comprising:

coupling a host channel adapter having a cache memory between a host processor and a switch fabric;

storing context information in a system memory associated with the host processor, the context information relating to a plurality of queue pairs for use in sending and receiving data packets over the fabric, each of the data packets belonging to a respective one of the queue pairs;

responsive to the use of a subset of the queue pairs to send and receive the data packets, loading into the cache memory the context information relating to the queue pairs in the subset; and

processing the data packets using the adapter based on the context information in the cache memory.

38. Data processing apparatus, comprising:

a multi-way set-associative cache memory, configured to hold multiple cache entries for use by a processor; and

a cache controller, coupled to receive a request by the processor to access a desired entry among the multiple entries in the cache memory, the request indicating an index of the desired entry and a target to

be matched by the desired entry, the cache controller being adapted, when none of the entries in the cache memory with the index of the desired entry matches the target, to block access to the entries in the cache memory that share the index of the desired entry while loading the desired entry into the cache from a main memory, and while allowing the processor to access the entries in the cache memory that have indices other than the index of the desired entry.

39. Apparatus according to claim 38, wherein the cache controller is adapted, before blocking the access to the entries in the cache memory that share the index of the desired entry, to complete servicing previous requests by the processor to access the entries in the cache memory that share the index of the desired entry.

40. Apparatus according to claim 38, wherein the cache controller is adapted, after loading the desired entry into the cache from the main memory, to provide the processor with the access to the desired entry and to unblock the access to the entries in the cache memory that share the index of the desired entry.

41. Apparatus according to claim 38, wherein the request by the processor has an instance number, and wherein the index comprises a predetermined portion of the instance number, while the target comprises another portion of the instance number.

42. Apparatus according to claim 41, wherein the instance number comprises a queue pair number.

43. Apparatus according to claim 38, wherein each of the entries in the cache has a respective flag that is set to

indicate that the entry is in use, and wherein the cache controller is adapted, upon loading the desired entry into the cache from the main memory, to store the desired entry in the cache in place of one of the entries that shares the index of the desired entry and whose flag is not set.

44. A method for data processing, comprising:

receiving a request to access a desired entry in a multi-way set-associative cache memory, configured to hold multiple cache entries, the request indicating an index of the desired entry and a target to be matched by the desired entry;

when the target matches one of the cache entries having the index of the desired entry, providing the requested access to the entry in the cache; and

when none of the entries in the cache memory with the index of the desired entry matches the target, blocking access to the entries in the cache memory that share the index of the desired entry while loading the desired entry into the cache from a main memory, and while allowing access to the entries in the cache memory that have indices other than the index of the desired entry.

45. A method according to claim 44, and comprising, before blocking the access to the entries in the cache memory that share the index of the desired entry, completing servicing of previous requests to access the entries in the cache memory that share the index of the desired entry.

46. A method according to claim 44, and comprising, after loading the desired entry into the cache from the

main memory, providing the access to the desired entry and unblocking the access to the entries in the cache memory that share the index of the desired entry.

47. A method according to claim 44, wherein the request by the processor has an instance number, and wherein the index comprises a predetermined portion of the instance number, while the target comprises another portion of the instance number.

48. A method according to claim 47, wherein the instance number comprises a queue pair number.

49. A method according to claim 44, wherein receiving the request comprises setting a respective flag for each of the entries in the cache to which the access is requested so as to indicate that the entry is in use, and wherein loading the desired entry into the cache comprises storing the desired entry in the cache in place of one of the entries that shares the index of the desired entry and whose flag is not set.